International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 274-295
https://doi.org/10.70917/ijcisim-2026-5431
Article

INTENT-GUIDED RETRIEVAL-AUGMENTED GENERATION WITH LORA-BASED INTENT ROUTING FOR INTELLIGENT CUSTOMER SUPPORT

Chethana T V[1], Dr. Payal Koolwal[2], Dr. R. Jennie Bharathi[3], Dr. Tamilarasan S[4], Dr. Rakhi Dua[5]

[1] Assistant Professor, Department of AI&ML, Bangalore Technological Institute, Bangalore. Email: chetugopal@gmail.com
[2] Professor, Department of CSE, Bangalore Technological Institute, Bangalore. Email: payalkoolwal11@gmail.com
[3] Assistant Professor, Department of CS & Business Systems, BMS Institute of Technology and Management, Bangalore. Email: jennie.bharathi@bmsit.in
[4] Professor, Department of CSE(AI&ML), AMC Engineering College, Bangalore. Email: elitetamilarasan74@gmail.com
[5] Assistant Professor, Department of Electrical and Electronics Engineering(SOET), K.R.Mangalam University, Gurugram. Email: rakhi.dua@krmangalam.edu.in

Abstract: Automated customer support technologies based on large-scale language models (LLMs) suffer from three classic failure modes: false creation of non-existing policies, inconsistent answers to the same questions, and misleading interpretations of user intentions due to confusing, joking, or vague inquiries. Large-scale language model (LLM)-based automated customer support technologies fail due to false policy creation, inconsistent answers to the same questions, and misleading interpretations of user intentions due to confusing, joking, or vague enquiries. Instant-Guided Retrieval-Augmented Generation with LoRA-based Intent Routing, a novel customer assistance system with intent-based retrieval, was successfully created. The Baseline pre-trained Llama-3-8B-Instruct model with no external grounding, a Naive Retrieval-Augmented Generation pipeline (Solution V1) that embeds SOP passages with sentence-transformers/all-MiniLM-L6-v2, indexes them in ChromaDB, and augments the prompt with the top-retrieved passage at inference time, and a Hybrid, The same externally held test partition was striated and an adversarial partition was screened for hedging and frustration language to evaluate all three procedures. Evaluations used ROUGE-1, ROUGE-L, BLEU, Format adherence rate, exact match, and fuzzy match accuracy. Based on 50 samples, retrieval increased ROUGE-1 from 0.2116 to 0.277, a 31.4% increase over the ungrounded benchmark, and BLEU from 0.0030 to 0.0328, an almost tenfold gain. With a fine-tuned intent router added to retrieval, ROUGE-1 would rise 3.9% to 0.2887 and BLEU to 0.0478 (+45.4% over V1). For the entire and adversarial sets, the intent router adheres to JSON Format 100% with 25-36% exact match and 42-46% fuzzy match accuracy.
Keywords: Retrieval-Augmented Generation (RAG); Large Language Models (LLMs); Intent Routing; Low-Rank Adaptation (LoRA); Dense Retrieval

1. Introduction

Conversational artificial intelligence is one of the applications most frequently used for commercial purposes, and customer support is one of the main functions this technology fulfills. The reason for the high importance of the implementation of sophisticated solutions in the field of customer support is that companies are required to process significant amounts of repetitive requests regarding refunds, delayed shipments, resetting passwords, disputes about bills, access to accounts, and technical malfunctions and, therefore, there is a definite need to automate basic levels of support by incorporating systems based on large language models (LLMs). The implementation of the LLMs is of particular interest because of their ability to come up with meaningful responses taking into consideration context without challenging training for the task at hand. However, the knowledge provided by LLMs is mainly parametric,

which means that such knowledge is implicit. Such knowledge does not mean that it is in accordance with the organizational policies and that it is constantly current. When it comes to customer support, there are certain disadvantages associated with such a limitation, such as inaccurate refund period, an eligibility condition that is not true, inaccurate escalation or a query that is sent to a wrong workflow.

Retrieval-Augmented Generation or RAG is a technique that has come forth as one of the ways of overcoming the shortcomings of pure parametric generation. In contrast to the previous techniques that worked only with the capabilities of a trained model, RAG incorporates elements of retrieval of information from an external knowledge source and making it available to the language model in the course of generating an answer (Lewis et al., 2020). The first RAG model was able to show the great potential of the combination of parametric generation capabilities and non-parametric external knowledge in terms of knowledge-heavy generation tasks. Other retrieval-based methods have proved to have an advantage of obtaining knowledge for language generation tasks that the model has no in its parameters (Guu et al., 2020). In a situation like this, it is a great advantage to have this ability since the policies, staff rules, work processes, technical procedures, and standard operating procedures of the organisation can be modified independent of a particular language model. Instead of having to retrain the model each time organisational knowledge needs updating, information stored externally can be modified, and relevant data retrieved automatically when necessary.

Though RAG improves factual grounding, the system can rely heavily on accurate and relevant retrieval. Traditionally, a RAG pipeline takes the original query and tries to match semantically similar content based on the original text's embeddings and its vector database representation. While this works well for simple and well-formulated questions, customer support queries are often informal, fragmented, complex, emotional, or vague. It's common for users to be informal, express frustration, uncertainty, or describe something using words and phrasing quite different from the organization's documents. This means two inquiries with very similar phrasing might be referring to completely different procedures, and conversely, the same intent could be described with radically different wording. Simply retrieving content based on Thera query could surface content that's topically relevant, but procedurally inappropriate. Therefore, the need arises for something that can find user's underlying intent prior to information retrieval.

An Intent-Guided RAG system solves this issue by adding an additional Intent-Routing layer to the existing RAG frame work: User Query → Intent-Routing → Information Retrieval (RAG). Instead of using Thera query, an intent router takes the query and converts it into a formal structured representation of the underlying user information need. This representation can then be used to inform the retrieval phase and direct the retrieval system towards the correct policy, SOP or knowledge domain. This becomes particularly pertinent in customer support, given how often different organizational documents may use overlapping terminology whilst differing in procedures, eligibility criteria, escalation paths or solutions. An explicit intent identification layer could thus provide the necessary signal for discriminating between those seemingly similar but fundamentally different requests.

Effective implementation of an intent-router does not necessarily demand the complete fine-tuning of a large model. Parameter-Efficient Fine-Tuning (PEFT) techniques offer an efficient alternative for fine-tuning a large language model (LLM) onto a narrow, specialised task without full model retraining. InLow-Rank Adaptation (LoRA) (Hu et al., 2022), for example, trained low-rank matrices are appended to certain existing transformer layers while most of the original LLM weights are frozen; this significantly reduces the number of trainable parameters. Lora (Dettmers et al., 2023) further optimises this approach by quantising the models used in conjunction with LoRA adaptation, enabling an efficient fine-tuning of LLMs for targeted applications even when computational resources are limited. These models are especially well-suited for tasks that require a narrow classification, or a structured prediction task targeting a limited behavioral capability. The present study, therefore, introduces an Intent-Guided Retrieval-Augmented Generation system with LoRA-based intent routing to facilitate domain-specific customer-

support response generation. The designed architecture leverages the complementary benefits of retrieval grounding and parameter-efficient task adaptation. The user's query is first fed into a LoRA-adapted intent router that disambiguates and classifies the utterance with the most pertinent support intent and generates a structured prediction.

The identified intent is then used to assist in the retrieval stage where relevant organisational documents/SOP paragraphs are identified. Finally, these documents serve as context to the language model, which subsequently generates the response to the customer's query. RAG here addresses the challenge of providing access to authoritative and updated organizational knowledge whereas LoRA-based intent routing addresses how this query is structured to aid query to document retrieval under the challenge of noisy and ambiguous queries. The Proposed Frame work is evaluated by comparing three models in an isolated manner: - An ungrounded Baseline, Naive RAG and Hybrid Intent-Guided RAG to individually establish the effect of retrieval grounding versus additional benefit of intent-guided routing. The generated responses are measured by Rouge-1 , Rouge-L , Bleau score and the intermediate routing component is evaluated by exact and fuzzy match accuracy and format compliance metric.

The dual aspect evaluation of responses and routing quality of both helps establishing the framework. The core rationale of this paper is that retrieval provides information needed to form responses that are authoritative and intent-routing aids in selecting information for retrieval, while using lean LoRA-based adaptivity for both intent selection and the main generation process. This approach to be explored contributes to retrieval augmented generation, parameter efficient fine tuning and intent-aware retrieving for building customer supported chatbot in domain specific areas.

1.2 Why the Problem Matters

A naive semantic retriever will simply embed the customer message directly and find the nearest policy chunk against that embeddign via a vector store query. This performs well if a query has surface surface structure aligned with underlying intent but the typical customer message often contains noise – hedging (“I guess”, “maybe”), negations (“still hasn’t arrived”, “can’t access”), and negative terms (“again”, “wrong”, “terrible”) that all contribute to embedding and may cause it to be dragged away from the correct policy into related but not equivalent document sections. If the policy retrieval step can’t successfully retrieve the underlying intent from the messy input, then the rest of the system might well be perfectly grounded on bad input. The retrieval step itself is therefore the bottleneck for some percentage of real-world queries and, for that reason, an additional upstream step is desirable – intent-normalization.

1.3 Limitations of Conventional Approaches

Purely parametric LLMs: Fluency and policy accuracy outstrip grounded knowledge; policy elements are decoupled from model training data and can only be updated through model retraining.

Rule-based or keyword-matching support bots: Predictable but brittle, fail on paraphrased, sarcastic or multi-intent queries, still need manual maintenance of rules.

Naive semantic search alone: Enhances grounding, but accuracy of retrieval deteriorates if the embedding is skewed, primarily due to the noise, hedging, or sentiment signals instead of the intent which prevails in actual customer messages.

1.4 Contributions

This paper makes the following contributions:

An end-to-end, reproducible pipeline that takes raw Bitext customer-interaction records and hand-authored, thirteen-document SOP knowledge base, and processes it via cleaning, deduplication, stratified partitioning, chunking, embedding, indexing, LoRA fine-tuning, and controlled, comparative evaluation.

A naive, Retrieval-Augmented Generation (RAG) pipeline (Solution V1) over the SOP corpus; uses sentence-transformers/all-MiniLM-L6-v2 embeddings indexed into ChromaDB; a system that stands on its own as a retrieval

evaluation mechanism, allowing the experimenter to control for all other factors when analyzing retrieval performance.

A LoRA fine-tuned structured intent router (Solution V2); it transforms noisy customer-level queries into strict JSON intent/category objects and then uses the resulting JSON object to construct targeted retrieval queries. In this pipeline, Solution V2's only difference compared to Solution V1's architecture is an appended retrieval system and the integration of an unmodified Solution V1 retrieval stage.

A controlled, three-way comparative evaluation of the Baseline, Solution V1 and Solution V2 across both identical held-out, and adversarial test splits (containing only utterances bearing a certain amount of hedging and frustration language) according to ROUGE-1/ROUGE-L scores, BLEU scores, Format Adherence, and Exact/Fuzzy-Match intent accuracy.

A data-leakage safe design; uses normalization based deduping followed by partitioning and employs set intersection checks to ensure no duplicate prompts are present within any partition.

The work is not intended to be a new retrieval algorithm, new fine tuning approach, or new base LLM architecture; the authors do not claim to obtain state of the art results against existing benchmarks. However, they present a solid, reproducible, and isolated experimental framework which measures the separate and combined impact of retrieval and fine-tuning over a realistic end-user customer support task.

1.5 Paper Organisation

In Section 2, we cover literature in retrieval-augmented generation (RAG), parameter-efficient fine-tuning, sentences embeddings, factuality of generation and intent classification. In Section 3, we formalise the problem setting. In Section 4, we introduce the proposed Intent-Guided RAG framework. In Section 5, we characterize the dataset used and detail initial exploratory analysis performed which guided pipeline design. In Section 6, we document process for each part of the pipeline, including LoRA configuration and training dynamics. In Section 7, we specify experimental settings and evaluation approach. In Section 8 we report quantitative results. In Section 9, we analyze result and explain observed trend as well as document the issues found and fixes in methodological execution along the development. In Section 10, we summarize our limitations, and in Section 11 we provide outline of future research.

2. Related Work

2.1 Retrieval-Augmented Generation

Lewis et al. (2020) proposed RAG (Retrieval-Augmented Generation) as a general-purpose fine-tuning recipe that couples a parametric generator with a differentiable non-parametric memory in the form of a dense vector index. They showed that RAG improves upon entirely parametric seq2seq baselines in knowledge-intensiveQA tasks and is significantly more faithful by producing more specific language. Closely related work, such as REALM (Guu et al., 2020), injects a learned retriever at training time so the LM can learn to retrieve and attend over external text to answer downstream questions. Recent surveys (Gao et al., 2023) show a swift evolution of RAG configurations - including naive, advanced, and modular varieties -and establish retrieval quality, not generation quality,as the major bottleneck with a decent baseline generator. This motivates us to focus on creating a better query for retrieval and not replace the generator.

2.2 Parameter-Efficient Fine-Tuning: LoRA and QLoRA

Full fine-tuning of an up-to-date instruction-tuned LLM is economically out of range for a specific narrow, single purpose adaptation task. Instead, Low-Rank Adaptation (LoRA) (Hu et al., 2022) locks the weight matrices of pre-trained models and inserts a pair of small low-rank decomposition matrices into certain linear layers in the self-attention module of selected layers – these are only trainable weights. This makes it possible to achieve comparable effectiveness of full tuning in numerous tasks with dramatically lower trainable parameters. QLoRA (Dettmers et al., 2023) optimizes LoRA by quantizing the fixed base model to 4 bits with Normal Float (NF4), dual quantizing the

quantisation factors, adding paged optimisers, which altogether renders the process available for personal machines or free Google-provided virtual machines (Colab / Kaggle). This entire area of study is summarised in Ding et al. (2023) under the banner of "delta-tuning" where they put LoRA and its variations into context among adapters, prompts and prefixes. In this project we use LoRA over 4-bits-quantized (NF4) Llama-3-8B-Instruct adapted for the very specialized task of structured intent extraction - based on the work mentioned earlier which shows even a few million trainable parameters are enough for such specific tasks.

2.3 Sentence Embeddings and Dense Retrieval

Dense retrieval relies on an embedding model to encode semantically similar text close in vector space. Sentence-BERT (Reimers & Gurevych, 2019) proposes a Siamese/ triplet fine-tuning objective over BERT style encoders aiming for sentence level embeddings that are comparable using cosine similarity instead of the quadratic-cost cross-encoding required by standard BERT, dropping similarity search compute for large amounts of text from tens of hours to seconds, while maintaining excellent performance on semantic textual similarity datasets. Our compact model, all-MiniLM-L6-v2, is a descendant from this line and is popular in production systems of RAG for the simple fact that it is the one with better balance between accuracy and inference speed for top-k similarity search of a small corpus.

2.4 Hallucination in Language Generation

Ji et al. (2023) present a survey on hallucination in natural language generation and discriminate intrinsic (the factual contents presented by the model directly conflict the sources) and extrinsic (content of the model is unsubstantiated and can't be verified by the sources) hallucination. As survey reports, hallucination is a commonly failure in summarisation, generation of dialogues, generative questions answering and grounded generation (referring to the retrieved factual contents which are verifiable) has shown to be among the most robust failure mode across these tasks, thus justifies Hallucination-adjacent grounding metrics- ROUGE and BLEU overlap compared to SOP-grounded references plus qualitative analysis as the major method used in this paper to determine whether retrieval and intent- guided routing can actually address factuality, instead of relying solely on model's self-evaluation or unshaped human evaluation.

2.5 Intent Classification in Task-Oriented Dialogue

The task of structured intent extraction is well established and widely used throughout the history of task-oriented spoken-language understanding. Among many prior results, Chen et al. (2019) was able to show that performing co-trained fine-tuning of BERT across intent classification and slot filling yields significant gains to both intent-classification accuracy and sentence-level semantic-frame accuracy over prior recurrent attentive-based approaches on two popular intent detection benchmark datasets (ATIS and Snips). Our approach uses a specific instance of this general concept – specialized, jointly trained, model for the sole task of generating a clean, structured semantic representation of user intent, but to a generative, decoder-only LLM by use of LoRA, integrated with an external retrieve search action, rather than a structured-fill approach inherent in slot filling approaches. This integration with a retrieval search rather than an in-pipeline fill is one of the particular adaptations this work provides relative to many works within the intent detection literature during the classification-based era.

2.6 Evaluation Metrics for Grounded Generation

Two common automated metrics reported to compare generated text with respect to a gold standard are BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), with the former being an n-gram overlap focusing on precision and introduced with machine translation, and the latter (in particular ROUGE-1 and ROUGE-L, related to unigram overlap and longest-common-subsequence overlap) more common for summarization and open-ended generation. Since both are indirect measures based on lexical overlap, and not direct measures of factuality, we follow common RAG practice, report them along with other (task specific) measures (Format Adherence, Exact Match, Fuzzy Match for the structured intent-routing) and don't use them as single measures.

Compared with these literature, the presented paper introduces no new retrieval algorithm nor new parameter efficient fine-tuning procedure, no new base models either. We show a controlled combination and comparison of a well known retrieval pipeline (Lewis et al., 2020; Reimers & Gurevych, 2019) and a well known PEFT procedure (Hu et al., 2022; Dettmers et al., 2023) on a narrow, well-defined intent-routing subtask, in the particular context of customer support, all assessed under a same fixed protocol applied in the same way for the 3 systems, so that any increase should be due to the method tested and not to some modified condition of evaluation.

3. Problem Statement

The current problem involves taking an unstructured customer support question and a collection of previously written corporate documents and utilising them to generate a response that (a) accurately identifies the customer's goal and the type of question asked, (b) is backed up by the firm's actual policy instead of data collected by the system, and (c) yields the same result regardless of how the question is rephrased.

3.1 Input and Desired Output

Input: The problem being a raw customer support query (free-text which could be messy, ambiguous, or sarcastic, or use hedging). Knowledge base = Corporated SOP Markdown docs. Topics: account recovery, billing dispute, data privacy, escalation, order tracking, password reset, payment method, product return, refund, shipping delays, subscription cancellation, technical troubleshooting, working hours.

Desired output: A structured JSON object containing the identified intent and category (rated on Format Adherence and Intent Accuracy), and a natural language response that uses retrieved SOP content (rated on the lexical alignment with SOP-grounded reference using ROUGE/BLEU).

3.2 Existing Limitations

The non- grounded generation can not assure relevance with real company policy.

Naive similarity search uses the full raw query as the retrieve signal, and is sensitive to noise, sentiment and surface phrasing rather than the original query intent.

Both alone, do not produce a machine-consumable, structured form of query intent that can be passed to downstream applications like ticketing, routing and analytics.

3.3 Technical Challenges

The task of cleaning and de-duplicating a real-world instruction set, while maintaining balance across partitions by instruction-class.

Chunking heterogeneous SOP Markdowns into retrievals which are neither so large that additional, nonrelevant text addsnoise,nor so small that exceptions and conditions drift away from their conditions.

Training a parameter-efficient adapter that outputs well-formed JSON within the bounds of constrained, low-bit-quantized, single GPU compute, and using its output directly in an existing retrieval interface without modifying the retrieval procedure.

Creating a hold-out split evaluation setting that provides identical and fair evaluation between the three systems, along with an adversarial subset.

4. Proposed System: Intent-Guided RAG with LoRA-Based Intent Routing

The system described below is called in all instances after Solution V2 or Hybrid RAG. It is defined as 3 successively improved systems that are judged using exactly the same procedure, so as to measure the utility of each one of them:

Baseline: Unmodified trained Llama-3-8B-Instruct creates a response based directly off of customer prompt with none of the context added via RAG and with no fine-tuning.

Solution V1 (Naive RAG): This raw customer query is encoded using sentence-transformers/all-MiniLM-L6-v2 and used to fetch the best matching SOP document (k = 1) out of the thirteen corporate policy documents that constitute the ChromaDB index and this retrieved SOP document is appended to the prompt.

Solution V2 (Hybrid, Intent-Guided RAG): V2: Lora-fine-tuned Llama-3-8B-Instruct intent router First translates the raw query string into a structured intent representation (JSON including Intent/Category), which then goes into mapping to a subject topic relevant retrieval string, feeding the same ChromaDB retrieval step as in V1, then feeds the same generation engine and decoding strategy as Baseline/V1.

This was by design, because with all modules after retrieval being fixed across V1 and V2 (embedding model, vector index, context assembly, prompt formulation and the generation model), the measured difference between V1 and V2 must be due to the quality of the retrieval query generated by the fine-tuned intent router.

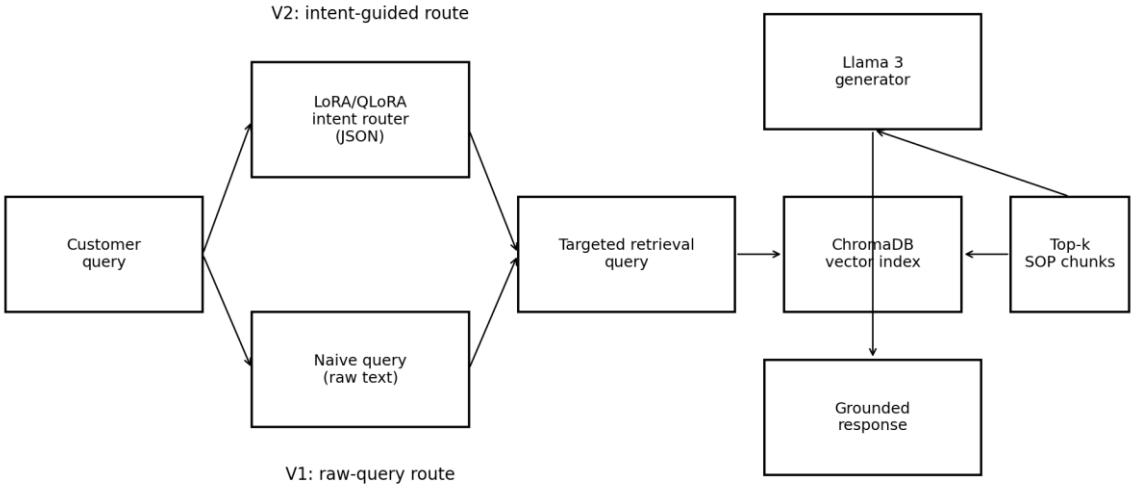

Figure 1. Intent-guided RAG architecture.

4.2 Component Overview

4.2.1 Generation and Routing Model -- Llama-3-8B-Instruct

In this setup, Llama 3 (Grattafiori et al., 2024) is used for both the generation and intent routing models, with the latter acting as a fine-tuned adaptor. Compatible with both prompt-based RAG and parameter-efficient adapter-fine tuning using the PEFT library, Llama 3 models are open-weight, instruction-tuned, decoder-only Transformers. Specifically, a single Tesla T4 GPU is used to fine-tune 8 billion Meta-Llama-3-8B-Instruct parameters (as loaded using Nous Research/Meta-Llama-3-8B-Instruct tokenizer/weights, guaranteeing repeatability).

4.2.2 Embedding Model -- sentence-transformers/all-MiniLM-L6-v2

This lightweight (Reimers & Gurevych, 2019) sent-embedding model, available in Python through the Hugging Face Embeddings library, encodes text into a dense, fixed-sizevectorspace with the characteristic property that two sentences are similar if their embeddings are close to each other in cosine space. The same model is applied to both SOP chunk indexing and the query encoding step: (otherwise the embedding comparison is not meaningful and you'd be mixing representations computed with different models).

4.2.3 Vector Store -- ChromaDB

ChromaDB can be used to store dense chunk embeddings, their associated metadata (source document, chunk number in doc) and provide rapid approximate nearest neighbour similarity search enabling us to fetch the "most relevant" SOP chunk for a particular query embedding linearly without scanning.

4.2.4 Intent Router -- LoRA-Adapted Llama-3-8B-Instruct

The intent router is a LoRA adapter (rank r = 16, alpha = 32, targetting the qproj and vproj attention projections, dropout 0.05) trained on the top of the 4 bit-quantised base model. Its single-purpose is very limited: transforming the raw user request to a json object labeling its intent and category. The resulting structured object is then only used to generate a vector search string targeting specific domains; it does not overwrite/bypass any part of retrieval or generation models.

4.3 Retrieval Mechanism, Context Augmentation, and Prompt Construction

Retrieval mechanism This process uses a very naive (not re-ranked) top-k similarity search to obtain the best k=1 scored chunk from ChromaDB index directly. Context augmentation combines the retrieved passage into a single chunk for grounding and prompt construction includes the customer query along with the context block and instructions for generating a response only using the context and for not making claims about details that were not found in it. This constitutes the fundamental way in which RAG is hypothesized to decrease hallucination versus the baseline - it requires the model to answer question based on retrieved and verifiable text.

4.4 Fine-Tuning Integration

The fine-tune stage is integrated as an upstream pre-processing layer rather than a replacement to some section of the RAG pipeline itself. So, in V2, the LoRA-trained router transforms the original customers' query into a typed JSON object representing the intent/category, converts such structured JSON object to string representation of search criteria (e.g. Map an extracted refund-type delay into "refund & shipping SOP string") then send it to unmodified Solution V1 retrieval mechanism (which remains the same embedding model, ChromaDB index, same chunking and prompt mechanism as Solution V1). This architecture also let us claim that "this chunk retrieval enhancement is specifically owed to the fine-tuned router, since the rest of the pipelines are identical".

5. Dataset and Exploratory Data Analysis

5.1 Data Sources

We developed the customer-support chatbots by drawing from two data sources: [1] the Bitext customer-support LLM chatbot training data (Bitext Innovations, 2023), an open-source instruction dataset for twenty-seven distinct customer support intents categorized into eleven intent groups (for example, ORDER, REFUND, and ACCOUNT) and with instruction, intent, category and reference response keys and [2] an internally drafted corporate SOP knowledge base of thirteen markdown documents representing account recovery, billing disputes, data privacy, escalation procedures, order information, password reset and updates, payment methods, returns, refunds, shipping delays, subscription cancellation, troubleshooting procedures and operating hours.

5.2 Dataset Characteristics

This “full” Bitext dataset has 26,872 instruction-intent-category-response entries split amongst 27 distinct intents and 11 distinct categories. Similar to common practice under constrained compute limits, we randomly (under strata, random_state=42) sub-sampled down to 4000 entries (avg of approx. 148 entries per intent). After searching the rows for nulls we confirmed that we have absolutely no missing data in any of the fields flag, instruction, category, intent, and response. Because near duplicate instructions (ie. Differing only by casing and spacing like “Where is my order?” vs “where is my order?”), standard exact-match de-duplication isn’t adequate for this dataset so, after de-duplicating using normalized instructions (lower-cased, whitespace-stripped, pre-splitting), we’ve removed 98 of

these rows yielding a cleaned-down data sample of 3902 entries. Examining label frequency at the intent level indicates no immediate concern with a severe imbalance-bias problem: the largest single label frequency category, review, occurs only on 4.3% of the cleaned data and, consequently, is relatively well-balanced which bodes well for a stratified splitting approach.

5.3 SOP Corpus Analysis

The TF-IDF vectorisation of the thirteen SOP documents (having filtered out stop words) yielded a 13 (documents)x 797 (terms) term-document matrix representing the 13 documents over 797 unique terms. Pairwise cosine similarity between each pair of SOP documents was computed with the goal of identifying redundant or conflicting policy statements which could create an ambiguous search scenario. With nearly every pairing appearing below 0.3, the similarity matrix demonstrated little general cross-document overlaps but did display a reasonable similarity value between account\recovery and password\reset of 0.42 (as might be expected given the nature of these SOP documents) as well as similarities between payment\methods, refund\policy and billing_disputes within a range of 0.26 to 0.28, these again indicate reasonable similarity and not actual policy text redundancy or overlap. There was no pair that passed a similarity threshold of 0.5 or greater which would flag it as a document pair that must be restudied and potentially restructured before chunking, thereby validating the decision to implement wholesale documents retrieval (k=1) in this use case instead of chunking on a finer level due to an overwhelming degree of policy text overlap or conflicting statements between SOP articles.

Table 1. Summary statistics from Stage-1 exploratory data analysis and Stage-2 data preparation.

Statistic	Value
Original Bitext dataset size	26,872 rows
Unique intents / categories	27 / 11
Stratified sample size	4,000 rows (random_state = 42)
Average examples per intent (sampled)	≈ 148.1
Null values found	0 (across all fields)
Near-duplicate rows removed (normalised dedup)	98
Final cleaned sample size	3,902 rows
Most frequent intent / share of sample	review / 4.3%
Corporate SOP documents	13
TF-IDF matrix shape (documents × terms)	13 × 797
Held-out test split (Stage 2 partitioning)	391 rows, stratified by intent

5.4 Data Preparation and Leakage Control

After performing EDA on the cleaned data, the sample was converted into ChatML format instruction examples with rigid JSON targets and tokenized with label masking. Label masking ensured that only the target JSON range was used to calculate loss (tokens which were not part of the target range were masked with -100). The sample was then split into training, validation and test sets by using stratified sampling on intent. Data leakage was checked by programmatically confirming (using set-intersection checks) that no prompt existed in more than one of the training, validation and test partitions and that the three partitions were statistically independent of each other prior to model training. The generated held-out test set available for final evaluation purposes also has 391 examples, stratified by intent, with no confirmed overlap with training and validation partitions.

6. Methodology

Our procedure uses a modular build-up, based on evaluation: We train a baseline without grounding, retrieve and add to the baseline to see how much effect retrieval alone provides, and we finally fine-tune with retrieval on top and look at the incremental effect. Everything is evaluated on the sameheld out data with the same deterministic decoding settings (do_sample = False, temperature = 0) so we know exactly what effect each component adds, without influencing other aspects of our experiment.

6.1 Stage 1 -- Baseline Model

Baseline Set with Unmodified Llama-3-8B-Instruct The base output from the raw Llama-3-8B-Instruct model exposed via Ollama (Q4 quantification). Model responses were generated based on the raw customer inputs and no prior retrieved context, no fine tuning. To ensure the responses are reproducible runs are deterministic temperature=0,topp=1, fixedseed). The goal here is to have a pre and post number to benchmark any further research work.

6.2 Stage 2 -- RAG Implementation (Solution V1)

The document ingestion pipeline is responsible for parsing the thirteen SOP documents and embedding each with sentence-transformers/all-MiniLM-L6-v2 before indexing them into ChromaDB, which we verify is built correctly by ensuring similar documents retrieve one another. For inference time, the same embedding model can be used to generate embeddings for the raw customer query. Similarity search against ChromaDB returns the most relevant document (k = 1) and this document, along with the user query and the instruction prompt to generate the final response are combined. In the strictest sense, retrieval is primitive in the sense that the similarity search takes the raw query from the user as input-there is no intent identification layer before query-time retrieval, which is the flaw that the fine-tuning from Stage 3 is specifically geared towards correcting.

6.3 Stage 3 -- LoRA Intent-Router Fine-Tuning

In this stage we configure and run parameter efficient fine tuning of the base model for the task of structured intent extraction. A 4-bit NF4 quantisation config (double quantisation, bfloat16 compute dtype) is applied to the Meta-Llama-3-8B-Instruct using BitsAndBytesConfig and the quantised model is made ready for k-bit training. We then apply a LoRA config (with target attention projections qproj, vproj, rank r = 16, alpha = 32, dropout = 0.05, bias = none) using the PEFT library, resulting in 6,815,744 trained parameters out of total of 8,037,076,992 parameters (trainable parameter fraction: 0.0848%).

We train on the label-masked ChatML-formatted instruction data from Stage 2 with AdamW (lr=210) at batch size 2 with a gradient accumulation of 2 (effective batch size=4) for a maximum of 500 steps, which is roughly 64% of one epoch across the ~3128 training rows (we observe 1561 batches per epoch with bz=2) . Validation loss is calculated on the validation loader every 10 steps and we apply early stopping on validation loss with patience=3 non-improving checks. We store the best checkpoint in terms of validation loss (not the loss of the last step).

The model was trained using one Kaggle Tesla T4 GPU, using the same random seed (42) throughout. The training was well and early stopped: validation loss was monotonically decreasing from 2.5917 at step 10 to 0.7034 where the training was stopped, for a final training step of 310 (not the specified max 500) triggered because validation loss has not decreased for the past 3 validation checks. Total wall-clock time for this run was ~4,824s (~ 80 mins). Table 2 shows the loss trend during the run for selected checkpoints.

Table 2. Representative train/validation loss trajectory during LoRA fine-tuning of the intent router (full log: 310 rows in training_log.csv, validated every 10 steps).

Step	Train Loss	Val Loss	New Best?
10	2.7017	2.5917	Yes
30	1.7875	1.8674	Yes

60	1.5325	1.5534	Yes
100	1.0753	1.2100	Yes
150	0.9770	0.8564	Yes
170	0.6448	0.8227	Yes
310 (early-stopped, best)	—	0.7034	Best checkpoint retained

Table 3. LoRA fine-tuning configuration and reproducibility information (from reproducibility info. json).

Parameter	Value
Base model	meta-llama/Meta-Llama-3-8B-Instruct (NousResearch tokenizer/weights)
Adaptation method	LoRA (r = 16, alpha = 32; target modules: q_proj, v_proj; dropout = 0.05)
Trainable parameters	6,815,744 of 8,037,076,992 total (0.0848%)
Quantisation	4-bit NF4, double quantisation, bfloat16 compute
Optimizer	AdamW, learning rate = 2×10^{-4}, constant schedule
Effective batch size	4 (batch = 2, gradient accumulation = 2)
Configured / actual training steps	500 (max) / 310 (early-stopped)
Validation cadence / patience	every 10 steps / patience = 3
Best validation loss	0.7034
Training environment	Kaggle, single Tesla T4 GPU
Total training time	≈ 4,824 s (≈ 80 minutes)
Random seed	42

6.4 Stage 4 -- Fine-Tuned RAG (Solution V2)

Instead of swapping the architecture out entirely, Solution V2 leverages fine-tuned LoRA adapter on top of the Solution V1 pipeline as an intent-routing pre-processing component. The fine-tuned model transforms raw customer query into intent and category JSON object, which we map to some specific focused vector search string then feed this string to the un-changed Solution V1 retrieval procedure. Both ends were confirmed by the integrated system before assessment, ensuring the router output is parsable as json and effectively drives the retrieval.

7. Experimental Setup and Evaluation Protocol

7.1 Systems Compared

Baseline -- Llama-3-8B-Instruct via Ollama, Q4 quantisation, no retrieval context, deterministic decoding.

Solution V1 (Naive RAG) -- raw query embedded with all-MiniLM-L6-v2, top-1 SOP document retrieved from a ChromaDB index of 13 policy documents, retrieved text appended to the prompt.

Solution V2 (Hybrid, Intent-Guided RAG) -- LoRA fine-tuned intent router extracts structured JSON intent, mapped to a topic-specific retrieval string driving the same ChromaDB retrieval step used in V1; final generation uses the same engine and settings as Baseline and V1.

7.2 Evaluation Data

In order to carry out the evaluation, a held-out, stratified test set that was equally distributed among all of the participants and contained 391 data was employed.For instance, we have made certain that all of the systems have an equal distribution, and we have prohibited any of the systems from leaking any data that they have seen previously during the test. This means that there are no actual overlaps with the train-valid sets. Additionally, the held-out test set was divided into a completely hostile subset, and with the help of regex, words that had a potentially unpleasant characteristic of speech were heuristically filtered away. This was done in order to ensure that the test set was completely different from the original. Negations, hedging language, and expressions of displeasure (such as "again," "cannot," "issue," "problem," and "wrong") are examples of phrases that fall into this category. On the basis of the test set, a subset consisting of fifty records was chosen at random (random_state=42) for the purpose of carrying out the comprehensive system comparison that is detailed in Section 8. For the purpose of determining the router's resistance, an extra twenty-four records, which together make up an adversarial subset, have been selected from the tests that have been held back.

7.3 Metrics

Table 4. Evaluation metrics used across all three systems.

Metric	What It Measures	Why It Is Relevant
Format Adherence Rate (%)	Whether the router's JSON output is syntactically valid.	A structured object that fails to parse cannot drive downstream retrieval or ticketing; this gates whether router output is usable at all.
Exact / Fuzzy Match (%)	Exact-match and fuzzy-match accuracy of the extracted intent against the labelled Bitext target.	Measures whether fine-tuning succeeded at its specific task of structured intent extraction.
ROUGE-1 / ROUGE-L	Unigram and longest-common-subsequence overlap between the generated response and an SOP-grounded reference answer.	A proxy for how closely the generated response's content matches an authoritative, policy-correct answer.
BLEU	Precision-oriented n-gram overlap with the reference answer.	A complementary lexical-similarity signal alongside ROUGE, standard in generation evaluation.

7.4 Comparison Methodology

Baseline vs Solution V1: isolates the effect of adding retrieval, holding the base model and evaluation set fixed.

Solution V1 vs Solution V2: isolates the effect of adding the fine-tuned intent router on top of retrieval, holding the retrieval architecture and evaluation set fixed.

Baseline vs Solution V2: reports the combined, end-to-end effect of both techniques together.

Held-out split vs adversarial subset: tests specifically whether performance degrades on noisy or sentiment-heavy queries, and whether fine-tuning narrows that gap relative to naive retrieval.

8. Results

8.1 Final Comparative Summary

Tabular representation of the final comparison analysis may be seen in Table 5. This assessment was carried out on a random sample of fifty records from the held-out test split (random_state = 42), with all three systems using the identical answer-generation settings. The random sample was chosen at random.

Table 5. Comparative ROUGE-1, ROUGE-L, and BLEU scores for Baseline, Naive RAG, and Hybrid RAG (n = 50, held-out test split).

System	ROUGE-1	ROUGE-L	BLEU
Baseline	0.2116	0.1089	0.0030
Naive RAG (Solution V1)	0.2779	0.1925	0.0328
Hybrid, Intent-Guided RAG (Solution V2)	0.2887	0.2014	0.0478

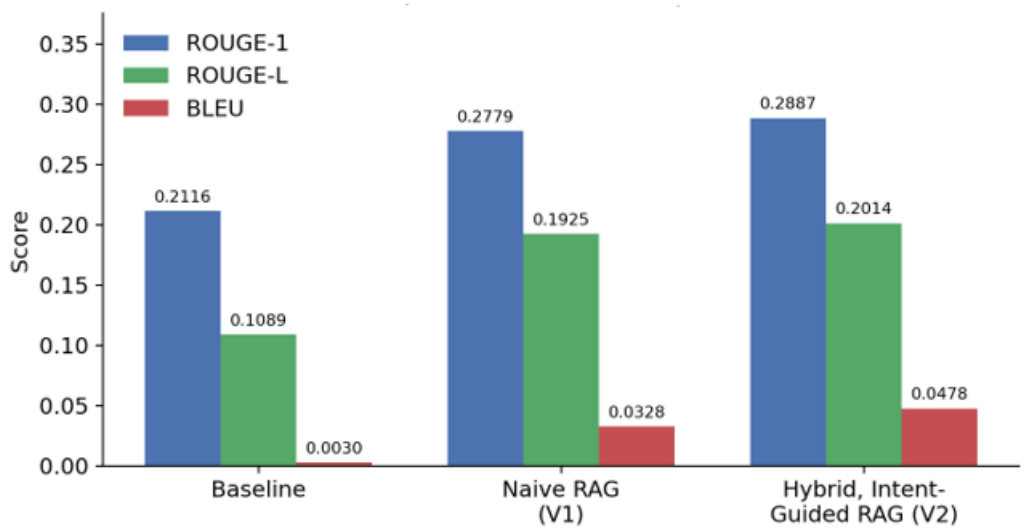

Figure 1. ROUGE-1, ROUGE-L, and BLEU by System (n = 50, held-out test split)

The ROUGE-1, ROUGE-L, and BLEU lexical-overlap measurements all show that the Baseline approach obtains lower values than the other two alternatives. Its lower performance in comparison to those of other options implies that generation without retrieval is not well linked with the replies that are most likely to be provided by customer support. In the pipeline, the step-jump boost in performance that is achieved by providing retrieval in Naive RAG (Solution V1) is by far the most significant.

The injection of pertinent documents from the SOP knowledgebase results in the generation of responses that exhibit higher levels of text- and sequence-level similarity in comparison to gold-standard answers. This would correspond to the significance of external knowledge grounding in the process of generating customer-support responses for situations in which organisational policies and procedures dictate the answer.

Since this is the case, Hybrid, Intent-Guided RAG (Solution V2) ultimately displays higher results on all three criteria. Given that it is superior than Naive RAG, it can be deduced that the extra LoRA-based intent-routing step possesses additional value in addition to the raw question retrieval stage. Before the system is able to pose a more refined inquiry, it must first recognise and frame the underlying customer intent. This allows the system to potentially get more relevant standard operating procedure background. Therefore, our findings demonstrate that there is a gradual enhancement throughout generational techniques, which may be classified as ungrounded generation, retrieval-grounded generation, and intent-guided retrieval-grounded generation depending on their respective classifications. In a summary, it would appear that intent-guided retrieval is able to enhance the generating outcomes more considerably than basic retrieval. However, the lexical overlap measurements are not able to show the factuality or demonstrate hallucination reduction merely by itself.

8.2 Improvement Attributable to Each Stage

Table 6. Relative improvement attributable to retrieval alone, fine-tuned intent routing on top of retrieval, and the combined effect.

Comparison	ROUGE-1	ROUGE-L	BLEU
Baseline → Naive RAG (retrieval)	+31.4%	+76.8%	+1008%
Naive RAG → Hybrid RAG (fine-tuning)	+3.9%	+4.6%	+45.4%
Baseline → Hybrid RAG (combined)	+36.5%	+85.0%	+1511%

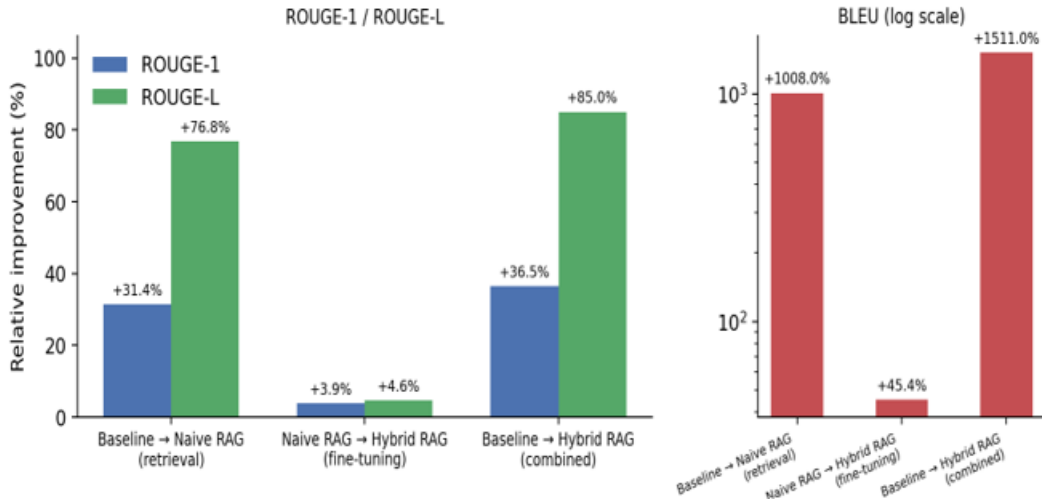

Figure 2. Relative Improvement by Comparison Stage

The most significant portion of the total improvement in response quality can be ascribed to the retrieval process. The difference between the ungrounded baseline and the Naive RAG (Solution V1) demonstrates a significant improvement on all three measures: a gain of 31.3% on ROUGE-1 (0.2116 to 0.2779), a gain of 76.8% on ROUGE-L (0.1089 to 0.1925), and a tenfold rise in BLEU (0.0030 to 0.0328). That the model is able to better root produced replies with goal outputs in the customer-support domain is supported by this observed result, which is consistent with the premise that the context is grounded in standard operating procedures. This considerable effect strength also lends support to our argument concerning the major flaw of Baseline, which is a lack of task-specific grounding and the possibility of lexical or factual drift rather than fluency in the language. The reason for this is because Baseline relies only on internal characteristics, which may or may not be reflective of the organization.

The improvement from Solution V1 to Solution V2, which is the contribution of the fine-tuned intent router, is significantly more mild on the ROUGE metrics as compared to the previous solution. 3.9% increase on ROUGE-1 (0.2779 to 0.2887) and 4.6% increase on ROUGE-L (0.1925 to 0.2014) are the results of the experiment. Having said that, the constant improvement across both ROUGE measures lends credence to the value of fine-tuned intent router. in the other hand, the rise in BLEU, which went from 0.0328 to 0.0478, is 45.7%, which indicates a more significant impact on n-gram level accuracy. It may be deduced from this that the intent-guided routing appears to be more successful in addressing the problems that arise when accurate and specific matches with retrieval results play a significant role and the underlying standard operating procedure that is retrieved is largely irrelevant. This is something that may occur when noisy customer inputs are present.

Overall, the investigations show that there is a pattern of growth that occurs in two stages: The retrieval process is responsible for the majority of the improvement; additional fine-tuning using a LoRA-based intent router permits targeted improvements, particularly in n-gram-based fluency, with a relative impact on BLEU that is more large than the impact on ROUGE measurements. These findings are consistent with the idea that the intent router makes it possible to repair some retrieval failures by having the capability to deal with errors that are brought about by query ambiguity or loud user inputs. Nevertheless, we acknowledge that this interpretation need to be seen as evidence-supported rather than definitive reasoning for retrieval correction, and that it ought to be verified with more retrieval-centric studies, if at all possible.

8.3 Router Evaluation

Separate evaluations were conducted on the whole held-out test sample and on the adversarial subset that contained hedging, denial, or frustration-related words. The format adherence, exact-match, and fuzzy-match accuracy of the fine-tuned intent router were examined individually.

Table 7. Fine-tuned intent-router performance on the full and adversarial evaluation subsets.

Subset	n	Format Adherence	Exact Match	Fuzzy Match
Full Test Set	50	100.0%	36.0%	42.0%
Adversarial Subset	24	100.0%	25.0%	45.8%

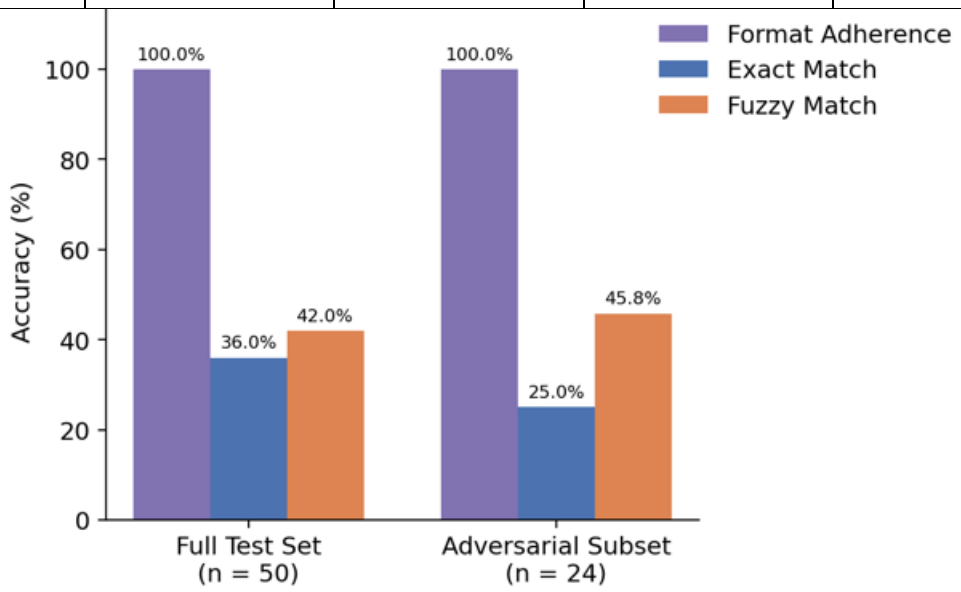

Figure 3. Intent Router Performance: Full vs. Adversarial Subset

Furthermore, Format Adherence is able to attain a perfect score of one hundred percent on both conventional and adversarial splits. This indicates that the intent router is able to generate outputs in the requisite parseable JSON format on a consistent basis when it is exposed to inputs that contain a certain level of noise or concealing. In addition, language that conveys displeasure is negative. As a result of valid structured output being the "low bar for entry" in terms of integration into an automated retrieval pipeline, the optimal format adherence outcome is an essential consideration.

A router that correctly anticipates the intended purpose but is unable to generate the parseable JSON that is requested would continue to be a source of discomfort further down the line and would be prohibitively difficult to employ as the foundation for retrievals. Therefore, a format adherence rate of one hundred percent on both subsets indicates that the fine-tuned router has acquired the necessary output schema and is able to keep it even when it is

exposed to a more hostile environment. The exact match performance of the router remains only moderate, ranging from 25% to 36%, which implies that the router does not always provide the exact intended purpose label. This is despite the fact that the router has a good performance on this structural measure. This constraint is made much clearer when one applies a fuzzy metric for accuracy to the label that was anticipated. The fuzzy match performance ranges from approximately 42.0% to 45.8%, which indicates that a significant portion of the predicted labels that were incorrectly attributed to exact intent classification are semantically close to the label that was intended when the fuzzy match was performed.

In actuality, this indicates that the intent router is able to recognise the overall operational environment, but it also has the ability to anticipate a label that is specific to the intent. Consequently, the disparity in performance between exact match and fuzzy match demonstrates the significance of assessing intent recognition on a number of different aspects. The exact match accuracy is a measurement that determines whether or not an identical label was created; nonetheless, it underestimates the actual utility of the router if it is one of the components of a retrieval pipeline. In conclusion, the findings suggest that there is a significant signal for the dependable structural output of router intent choices, as well as a modest measure of semantic correctness. Furthermore, it is recommended that fuzzy and retrieval-level measurements be utilised in conjunction with exactmatch accuracy techniques.

9. Discussion

9.1 Interpretation of Results

The individual contributions to grounding and routing are discussed in 8.2.1 As seen in Sections 8.1 and 8.2, retrieval grounding, which incorporates documentation that is particular to the organization, is the key contributor to the overall performance gain of the system. Fine-tuning intent routing, on the other hand, provides an additional performance boost, although a more specialised one.1 More specifically, the Naive RAG model (big ROUGE-1, ROUGE-L, and BLEU uplift) is preferred over the ungrounded Base model, which results in a considerable increase in the overall performance of the system. The fact that this is the case shows that the 8-billion parameter, instruction fine-tuned generation model of the Base model is typically powerful enough for general language production. When it comes to the support environment of an organization, however, fluent replies that are generated without access to specific rules, processes, and standard operating procedures are not sufficient. On account of this, the limitation of the Base model for such a context was the absence of the organization's own knowledge, rather than the capacity to provide generic information. It is the efficient retrieval of organisational information that is the source of the benefits that Naive RAG features. This provides an explanation for the primary contribution that retrieval made to the optimisation of the system as a whole.

In addition, we have noticed that the fine-tuning of purpose routing brings about further improvements in comparison to Naive RAG, with consistent benefits across all of the metrics that are included. However, the improvement that can be attributed to intent routing is not as significant as the initial benefit that may be attained by retrieval. In light of this, it can be deduced that the purpose routing functions more as an optimisation over retrieval than as a substitute for the fundamental information retrieval capabilities. A disproportionate amount of value appears to accrue to those searches in which the straightforward direct ungrounded embedding retrieval may only produce a document that is very broadly related but does not include the precise standard operating procedure and policy. The fact that the BLEU metric improves proportionally more is a strong indicator that the routing helps identify not only passages that are semantically related, but also those passages that are most relevant to obtaining the precise information that is required for the expected answer output. This is in contrast to ROUGE-1, which measures single word overlaps and generally benefits from both topical relevance and good phrase alignment.

For the sake of a brief summary, the suggested enhancements can be generally categorised into two main stages. To begin, retrieval grounding is a process that roots the model with the unique context of the organization, which ultimately results in significant gains in performance. By ensuring that the documents that are obtained are relevant and precise at the query level, fine-tuned intent routing then serves to optimise the documents that are retrieved, which

ultimately results in better aligned replies. The Hybrid, Intent-Guided RAG is an approach that has the potential to be useful since it combines retrieval augmentation with selective query optimisation.

9.2 Debugging Findings and Methodological Corrections

Following the completion of a systematic debugging procedure, as well as methodological validation and the correction of abnormalities that surfaced during the phases of development and assessment, the conclusive results that are summarised in Table 5 were finally achieved. Due to the fact that these anomalies had a significant impact on the results that were drawn from the comparison tests, it is absolutely necessary that they be documented. It was determined through preliminary assessment that the fine-tuned intent-routing approach was functioning better than the Naive RAG system. This evaluation was carried out at the beginning phases of the tests. It is possible that these findings could have been considered conclusive without further investigation, which would have resulted in the incorrect conclusion that intent routing was detrimental for response retrieval and generation. However, this conclusion could only be reached after it was determined that the early discrepancies in experiment results were partially due to evaluation and system integration confounds, as will be discussed in the following paragraphs.

During the process of debugging, there was a source of the difficulties that were detected, and that cause was inconsistency in the application of response generating engines and decoding protocols throughout specific intermediate trials. Variability in the system setup influenced their effect on the response generated, which resulted in discrepancies between their lexical overlap scores and the corresponding ground truth responses. This was due to the fact that modifications in the parameters set for decoding have a significant impact on both the choice of word and overall syntax, which in turn alters sentence composition. Despite the fact that the content of the documents retrieved by the system and user queries may have been consistent, the variance in the system setup was responsible for the effect. Therefore, a direct comparison between two systems that were assessed using different generation configuration protocols is not able to correctly reflect the function of either the routing component or the retrieval component.

A comprehensive overhaul of the experimental apparatus has been carried out as a solution to the problem. All three systems, namely Baseline, Solution V1 (Naive RAG), and Solution V2 (Hybrid, Intent-Guided RAG), have been tested by employing the same response generation engine that was mentioned before. Additionally, the decoding methodologies that are given in Section 7.1 have been followed in the same manner. For the goal of ensuring that the experiment as a whole is more reliable and consistent, all of the experiments have been carried out using the system settings that have been described. These configurations include Ollama and Llama 3, with temperature set to zero, top_p set to one, and seed set to a constant value. All of the variations in the replies can thus be traced completely to the modifications that were made to either the routing or the retrieval approach after these settings have been applied. After this modification had been made, the tests that had been carried out were re-executed using the identical assessment technique that had been used previously. By comparing Table 5, which demonstrates that there was a consistent and unequivocal ascent from the Baseline system to the Naive RAG to the Hybrid RAG for Rouge1, RougeL, and Bleu, it is possible to simply draw the conclusion that this particular progression occurred. As a consequence, the experimental configuration that was ultimately chosen offers a more definitive experimental framework for assessment than the preliminary ones, which produced contradictory results when compared.

The process for debugging and the findings offer light on the problem of experimental reproducibility and validation in the field of machine learning and research and development for Retr-Aug Gen. It is possible for the experiment to be significantly influenced by a variety of factors, including variations in the prompt, variations in the model version, variations in the decode process, the seed that is utilised, and the retrieve circumstances. If none of these factors are maintained as standards, it is possible that the differences in outcomes across techniques might be misinterpreted. In our opinion, the initial regressing outcome that was indicated before provided a clear illustration of this issue when we were carrying out the task.
It is for this reason that the utilisation of a single coherent evaluation mechanism that remains constant throughout the execution of each experiment is a more reliable option. This is due to the fact that all experimental configurations, procedures for testbed, answer generation protocols, and evaluation measurement systems that are utilised to evaluate

each methodology that is being examined should, ideally, yield compatible conclusions when comparing different systems. As a result, a more in-depth examination of the table reveals that it unequivocally demonstrates the increase in all three Rouges that occurs with each addition to the baseline approach (retrieval, followed by purpose routing).

9.3 Error Analysis

In order to find forms of failures that were not completely exposed by the aggregate metrics alone, a qualitative evaluation of both the output of the router and the answers made by humans was also carried out. The quantitative evaluation does provide a summary appraisal of performance; however, it does not provide any hints as to why any particular prediction is made wrong or how that inaccuracy may effect the retrieval and production process that occurs further down the line. Within the scope of the qualitative analysis, two aspects of the router's predictions were specifically investigated: the structure of these forecasts, as well as the characteristics of misclassifications.

The first observation that is worthy of remark is the discovery that the performance of Format Adherence was 100% for both the whole test and hostile test subsets. This indicates that the router created well-formed JSON in a systematic manner, even when provided with inputs that were difficult to understand or intentionally meant to do harm. Due to this rationale, it is highly improbable that problems in the JSON format and parsing led to the "remaining" categorisation mistake that was reported in the evaluation summary. In light of the fact that the router output is intended to function as a "machine-readable intermediate layer," this is an essential problem to consider while designing the system. The fact that the router and the retrieval system have robust interfaces despite severe language variance is proven by the high score for format adherence. It is suggested by the explanation that the distinction between the two performance levels is the more significant problem that needs to be addressed.

The accuracy of the full test set was around ten percent different between the stringent and fuzzy matches (25 to 36 percent for the strict match and 42 to 45.8 percent for the fuzzy match). As a result, many outputs that have been labelled "incorrectly" are not entirely irrelevant to the genuine intention that lies under the surface. Instead, it appears that many of them are within the "neighbourhood" of the accurate prediction in terms of the semantic or operational similarity they have with the proper standard operating procedure. This might comprise two distinct aspects of a same standard operating procedure (SOP), two separate phases in a given operation, or various sorts of assistance requests made by workers that are similar to one another. Furthermore, despite the fact that such inputs are given a poor grade according to a rigid criterion, they frequently include information that will be slightly or very useful to the retrieval process further down the line. It is very helpful to comprehend the impact of modifying the routing system once this comparison is taken into consideration.

The results of the full test set demonstrate that tuning intent routing brings a moderate benefit to the overall response of the system. Furthermore, the results also demonstrate where a significant portion of that benefit lies. The error analyses highlight how improved performance of the router at the routing layer concentrates a moderate impact rather than a uniform improvement throughout the performance of the entire system. The comparisons of the tests are also extremely essential since the difference in precise performance on the entire and adversarial test sets provides information that is meaningful. In the entire test setting, this figure was 36%, however on the hostile test set, it dropped to 25%. Additionally, the value on the complete test setting was 36%. When adopting a variant of input queries with increased complexity, the exact accuracy definitely declines, but at the same time, the fuzzy accuracy increases from 42 percent on the entire set to 45.8 percent on the adversarially constructed. The fact that variations such as hedging or doubt, emotion-bearing and frustration-oriented language add to linguistic complexities and might potentially distort the real procedure's aim is one possible explanation. It is therefore possible for the user to address the very same issue, but it is possible for him to use language that are indirect, that make him feel some annoyance, or even that communicate worries regarding many issues all at once.

It was possible for the router to correctly identify a generic subject, but it would assign it to a SOP index that was similar to it rather than one that was identical to it. A behaviour like this would appear to be more compatible with ambiguity or a lack of finer distinctions between existing related SOP types than it would be with a total routing failure. As a result, the primary issue that the system that is currently being presented is facing is not the generation

of JSON that is structurally sound or poor performance under diverse inputs. Rather, the primary issue is the ambiguity that exists between entities that are similar, for which even very good performance is anticipated to yield, if nothing else, relevant retrieval performance. Based on the fact that the format adherence tests were performed perfectly and that the performance on a relaxed metric was significantly better, it appears that the mistakes are more frequently located within connected regions rather than in parts that are wholly unrelated. Therefore, the next step in improving this system should not consist of concentrating on any enhancements for the JSON creation. Instead, the focus should be on defining key SOP categories and providing a variety of instances of ambivalent, multi-label, and frustration-oriented enquiries in the process of fine-tuning the router.

9.4 Comparison to Related Work

According to the existing RAG research, the magnitude and type of the performance benefits that were discovered in this investigation are mostly compatible with one another. This finding lends credence to the overarching idea that grounding a powerful language model in the retrieved, authoritative information that is relevant to a domain enhances the factualness and lexical coverage of its output. The largest substantial gain that was discovered was between the ungrounded Base model and the Naive RAG. Lewis et al. (2020) initially proposed RAG as a system that combines parametric language generation with non-parametric knowledge retrieval. They demonstrated how retrieving knowledge that is relevant to a query could be more beneficial than simply relying on language model parameters that encode world knowledge in a general sense. RAG was initially proposed as a system. Since then, more research has reaffirmed the need of grounding a produced response in some external information when it is appropriate. This may contribute to a reduction in the development of material that is either unsupported or over-hallucinated in situations that require a significant amount of knowledge (Ji et al., 2023).

Rather than training data on the vast amount of general-purpose language, the case of the organisational support environment that was studied here highlights this concept in a different way. The information that is most relevant to a request is the information that is pertinent to our organization's procedures, policies, and standard operating procedures (SOPs). As a result, the significant disparity between Baseline and Naive RAG highlights the significance of gaining access to authoritative organisational content rather than merely increasing language generation capability in general. This is of utmost importance in the context of support, where information that is fluently misinformed can sometimes be just as detrimental as information that completely misses the point if the procedures that are provided are not adhered to.

It is possible that the comparatively modest incremental benefit that results from precisely adjusted intent routing is representative of the nature of parameter-efficient optimising: LoRA was initially developed by Hu et al. (2022) as a technique for adapting large, pre-trained language models without modifying their entire set of parameters. This was accomplished while adapting models in other ways (for example, in some way on specific tasks). Subsequent works have further expanded on parameter-efficient tuning for large language models. This effort included the modification of roughly 6.8 million parameters out of a total of around 8 billion parameters in the model. This is approximately 0.0848% of the total parameters in the model, and it led to a statistically significant improvement in downstream measures. If the task can be modelled within a specialised, limited space of parameters, then a big pre-trained language model may be adapted to tasks without the whole parameter set being modified. This is consistent with the research, which shows that this is possible. Due to the fact that the current study modelled an intent routing job, which is by its very nature in a confined parameter space and has an operational purpose, it lends itself well to a format that can be tuned effectively.

At the same time, the fact that a more modest enhancement was achieved by parameter-efficient tuning demonstrates that it should not be confused with a replacement of the main RAG gain. In the context of open-ended generation, access to organisational content continues to be the foundational component of an appropriate answer. On the other hand, parameter-efficient tuning primarily assists by correctly navigating to the appropriate organisational content through the routing mechanism. Rather than competing against one another, RAG and parameter-efficient tuning complement and enhance each other. Overall, this work demonstrates that the findings of previous research are

supported by experimental evidence and that it is beneficial to incorporate grounding with lightweight task-specific intent tuning into an organisational support setting. It also demonstrates that significant improvement gains can be achieved even without the need to fully fine-tune an 8-billion parameter model.

10. Limitations

Evaluation Scale: The controlled evaluation referred to in Section 8 was carried out on a sample consisting of 50 records out of 391 in the held-out part of the dataset (plus 24 samples belonging to the adversarial set), which is important in terms of making comparisons in situations where monitoring of resources is important. However, it is not enough for making claims about production readiness; results should be considered reliable, but not accurate population estimates.

Retrieval depth: Retrieval is executed just for the single document that received the highest score (k = 1) instead of the multi-passage context that is re-ranked and makes attribution of the V1 and V2 comparison easier but may understate the upper limit of retrieval the advantages of which can be accessed with deeper retrieval.

Fine-tuning budget: The LoRA router model went through 310 steps out of total 500 configured ones (≈ 64% of 1 epoch on ≈ 3,128 rows) before the stopping of training processes and was trained on a single T4 CPU; it is clear that the increase of the budget could change the magnitude of improvements introduced by V1 and V2 with the expected results expected.

Base-model substitution risk: the proposal predicted that hardware limitations would lead us to substitute a smaller instruction-tuned model for Llama-3-8B-Instruct; although this would not turn out to apply in the final experiments reported here, the effect remains relevant and should be taken into account in case of further studies conducted on lower-spec machines.

Lexical-overlap metrics as a hallucination proxy: ROUGE and BLEU metrics are not directly related to factuality measures; nonetheless, they are commonly accepted as factually relevant proxies in academic studies (Ji et al., 2023) providing one with a certain range of readings of factual correctness.

Constructing an adversarial subset: the adversarial subset is formed through regex filtering applying a set of words that indicate hedging and frustration; the approach enables selecting interesting and useful examples of non-standard usage of language but does not cover all the possibilities of ambiguous or sarcastic customer speech.

11. Conclusion

The proposal predicted that hardware limitations would lead us to substitute a smaller instruction-tuned model for Llama-3-8B-Instruct; although this would not turn out to apply in the final experiments reported here, the effect remains relevant and should be taken into account in case of further studies conducted on lower-spec machines. Lexical-overlap metrics as a hallucination proxy: ROUGE and BLEU metrics are not directly related to factuality measures; nonetheless, they are commonly accepted as factually relevant proxies in academic studies (Ji et al., 2023) providing one with a certain range of readings of factual correctness. Constructing an adversarial subset: the adversarial subset is formed through regex filtering applying a set of words that indicate hedging and frustration; the approach enables selecting interesting and useful examples of non-standard usage of language but does not cover all the possibilities of ambiguous or sarcastic customer speech. Future research might enhance this comparison through multiple approaches: testing deeper, re-ranked retrieval systems (not just one document retrieval); refining the LoRA routing mechanism to include more alternates – potentially checking if V1 and V2 continue to differ with more fine-tuning coverage; challenging different subsets of queries to reach beyond keywords and include queries that AI systems determine to be ambiguous or sarcastic; carrying out a more rigorous evaluation of hallucination and answer correctness.

REFERENCES

1. Bitext Innovations. (2023). Bitext customer support LLM chatbot training dataset. Hugging Face. https://huggingface.co/datasets/bitext/Bitext-customer-support-llm-chatbot-training-dataset
2. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.
3. Chen, Q., Zhuo, Z., & Wang, W. (2019). BERT for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909.
4. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. Advances in Neural Information Processing Systems, 36, 10088-10115.
5. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems, 36*, 10088–10115.
6. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1*, 4171–4186. https://doi.org/10.18653/v1/N19-1423
7. Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., Yi, J., Zhao, W., Wang, X., Liu, Z., Zheng, H.-T., Chen, J., Liu, Y., Tang, J., Li, J., & Sun, M. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence, 5(3), 220-235.
8. Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., Yi, J., Zhao, W., Wang, X., & Sun, M. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence, 5*, 220–235.
9. Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2023). RAGAS: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.
10. Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., & Li, Q. (2024). A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. *Proceedings/Preprint*. https://arxiv.org/abs/2405.06211
11. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.
12. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.
13. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., et al. (Llama Team, AI @ Meta). (2024). The Llama 3 herd of models. arXiv preprint arXiv:2407.21783.
14. Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). Retrieval augmented language model pre-training. Proceedings of the 37th International Conference on Machine Learning (PMLR), 3929-3938.
15. Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). REALM: Retrieval-augmented language model pre-training. *Proceedings of the 37th International Conference on Machine Learning, 119*, 3929–3938.
16. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. Proceedings of the International Conference on Learning Representations (ICLR).
17. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*. https://arxiv.org/abs/2106.09685
18. Huang, Y., & Huang, J. (2024). A survey on retrieval-augmented text generation for large language models. *arXiv preprint arXiv:2404.10981*.
19. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), Article 248, 1-38.
20. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12), Article 248, 1–38. https://doi.org/10.1145/3571730
21. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020). Dense passage retrieval for open-domain question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550
22. Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 39–48. https://doi.org/10.1145/3397271.3401075
23. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.
24. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474.
25. Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, 74-81.

26. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
27. Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 311-318.
28. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3982-3992.
29. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3982–3992. https://doi.org/10.18653/v1/D19-1410
30. Saad-Falcon, J., Khattab, O., Potts, C., & Zaharia, M. (2024). ARES: An automated evaluation framework for retrieval-augmented generation systems. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1*, 338–354. https://doi.org/10.18653/v1/2024.naacl-long.20
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.
32. Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A. S., Arias, L., Bhatt, G., ... Khashabi, D. (2022). Benchmarking generalization via in-context instructions on 1,600+ language tasks. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1–27.
33. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 38-45.
34. Zhao, S., Yang, Y., Wang, Z., He, Z., Qiu, L. K., & Qiu, L. (2024). Retrieval augmented generation (RAG) and beyond: A comprehensive survey on how to make your LLMs use external data more wisely. *Microsoft Research*.